Homework 3 Solution

CS 1323, Fall 2015

1. (10 points; 5 points each) Use memory diagrams to trace the code below:

a)

int[] data;

data = new int[5];

for (int i=0; i<data.length; ++i)

data[i] = (i+1)\*3; Heap

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Identifier** | **Address** | **Contents** |
|  |  | 1000 | 3 |
|  |  | 1001 | 6 |
|  |  | 1002 | 9 |
|  |  | 1003 | 12 |
|  |  | 1004 | 15 |
|  |  | 1005 |  |
|  |  | 1006 |  |

Stack Frame

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Identifier** | **Address** | **Contents** |
|  | data | 100 | 1000 |
|  |  | 101 |  |
|  |  | 102 |  |

b)

int[] data = {1, 3, 5, 7, 9};

int[] copy = data;

copy[0] = 11;

copy[1] = 9;

Heap

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Identifier** | **Address** | **Contents** |
|  |  | 1000 | ~~1~~  11 |
|  |  | 1001 | ~~3~~ 9 |
|  |  | 1002 | 5 |
|  |  | 1003 | 7 |
|  |  | 1004 | 9 |
|  |  | 1005 |  |
|  |  | 1006 |  |

Stack Frame

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Identifier** | **Address** | **Contents** |
|  | data | 100 | 1000 |
|  | copy | 101 | 1000 |
|  |  | 102 |  |

1. (30 points; 10 points each) Trace the following for loops using the table on the right. Show every time a variable is changed—including the last change. If the code is illegal or does not run properly, trace as far as you can.

|  |  |
| --- | --- |
| **sum** | **count** |
| 0 | 0 |
| 1 | 1 |
| 10 | 2 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

int[] data = {1, 9, 7, 4, 2}; // constructs and initializes an array

int sum = 0;

for (int count = 0; count < data.length/2; ++count)

{

sum = sum + data[count];

}



|  |  |
| --- | --- |
| **sum** | **index** |
| 0 | 6 |
| 2 | 5 |
| 3 | 4 |
| 9 | 3 |
| 10 | 2 |
| 11 | 1 |
| 15 | 0 |

int[] data = {9, 4, 1, 7, 6, 3, 2};

int sum = 0;

for (int index = data.length-1; index >0 ; --index)

{

if (data[index]%2==0)

sum = sum + data[index];

else

sum = sum + 1;

}

* 1. int[] data = {10, 8, 6, 4, 3, 1};

|  |  |
| --- | --- |
| **sum** | **index** |
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| Program fails, index out of bounds | 6 |
|  |  |
|  |  |

int target = 5;

int sum = 0;

for (int index = 0; index < data.length && data[index] != target || data[index] !=target+2;

++index)

{

sum = sum + 1;

}

1. (5 points; 1 point each) Find the value of each of the logical operators below. If the statement is illegal in Java, say so.
   1. 8 < 12 && 15 < 7

false

* 1. 8 < 12 || 15 < 7

true

* 1. 14 < 9 || 12 != 5 && 3 == 9

false || true && false

false || false

false

* 1. !4<9 && !3 > 7

Illegal in Java, ! has higher precedence than <, and !4 doesn’t mean anything since 4 isn’t a logical value

* 1. ! (3 != 7) && 4 == 2

! true && false

false && false

false